DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER

July Number 23 1977

Contributions and correspondence should be sent to:

Robert Hassinger, Coordinator
12 Bit SIG
c/o DECUS
146 Main Street
Maynard, MA 01754

..or..

Liberty Mutual Research Center
71 Frankland Road
Hopkinton, MA 01748

DECUS/Europe contributions are solicited through:

Lars Palmer
DECUS/Europe 12 Bit SIG Newsletter Liaison
Hassle
Fack
S-431 20 MOLNDAL 1
SWEDEN

(Please include reference to Newsletter number and page when inquiring about material published.)

NEWSLETTER DEADLINE

The deadlines for ready-to-use material for the next Newsletter is 26-August-1977. Material requiring editing/re-typing must be in earlier. Ready-to-use material should use an area 6 1/2 inches (16.5 cm) wide by no more than 9 inches (23 cm) long on each page. It should be single spaced on white bond paper whenever possible and must be reasonably clean, legible and sufficiently dark for good photographic reproduction.

OS/8 V3D

At the Spring Symposium DEC confirmed plans for OS/8 V3D. This will be basically a "maintenance" release with normal bug fixes plus the solution to the "date problem" that was documented in previous Newsletters. The principle enhancements are only minor things mostly for the sake of the VT78 and OS/78. Full details were not available. The new release will have to include updates to the basic system kit, the extensions kit, and the FORTRAN IV kit because the date fix impacts them all. A firm date for the release was not given but for practical reasons it has to be early Fall. Everyone's dates will stop working on the first of January so they all have to have gotten an update by then if they want the date to keep working. The pricing policies seemed uncertain. DEC recognized the problem that the need to make the date fix represents so that it would be nice to make the updates available at the lowest possible cost. On the other hand DEC seldom passes up a chance to make money so I have no idea what will happen and at last report the Product Line did not seem to know eather.

MACREL

I think that DEC confirmed a commitment to release MACREL at the Spring Symposium (I hesitate to say for sure because there have been questions on this point in the past). Several field test versions have been run by a number of sites this Spring. Some of the sites reported on their experiences at the Symposium. Most of the reports were to the effect that they had (as expected) found a fair number of bugs but that overall they were enthusiastic about the basic design and progress to date. At the moment I think the final decision to release what they have is still pending but the momentum is towards getting a first version out even if it is not quite perfect or complete. If that decision is made soon, the first release could be available as soon as the middle of the Fall, however, I don't think any commitment to a date exists yet. As reported in the DECstation 78 article, a price of $350 was mentioned but I have no idea if that is official.

DECstation 78

At the Spring Symposium, DEC announced the DECstation 78, a sort of an LSI PDP-8 with video console terminal and floppy disks. DEC has not chosen to make available details on this product for the Newsletter so I will have to depend on my own recollections and notes from the Symposium and the limited information in their sales brochure. I trust DEC will correct any resulting errors in the next Newsletter.

The heart of the DECstation 78 is the VT78. It is a VT52 terminal in its normal enclosure with the addition of one large circuit board in the bottom that mounts a complete LSI PDP-8 based on the IM6100 microprocessor chip. Included on the board are 16K of memory with control, a real-time clock, asynchronous communications, disk and high speed data interfaces, "electronic program injection" (EPI), and ROM for control in place of a front panel. The VT52 and the processor are independent with a standard serial interface between them that makes the VT52 look like the normal console terminal. This interface runs at up to 9600 baud. The DECstation package includes a dual floppy disk (RX78) that should be compitable with the present RX01.

The I/O panel on the back of the processor contains five ports. Two are serial EIA RS232C ports that can be programmed for speeds of 50 to 19,200 baud. There is a parallel I/O port for printers and custom interfacing that provides bidirectional 12 bit transfers at up to 15K words per second (i.e. programmed transfer - there is no Data Break/DMA available or possible). A disk interface port allows connection to the RX78. Either one or two dual drive units can be connected for a total of either two or four drives. The fifth port is for the EPI. This is a box that plugs into the connector which contains ROM chips organized like a binary paper tape. Part of the built-in CPU control ROM contains a version of the binary paper tape loader arranged so it can read the contents of the EPI box into memory. This gives the possibility of loading diagnostics or stand alone programs on systems without a disk.

It can be arranged to automatically load memory from the EPI box each time the system is started for example.

DEC also announced the LA78 DECprinter. It seems to be an LA180 (180 character per second serial dot matrix printer) that can plug into one of the interface ports. They also showed the LPQ78 "letter quality printer" (i.e. standard "daisy wheel" type printer) like the one they have been putting into their Word Processing Systems. It interfaces through the 12 bit port to give full control of the printer's modes.

The OS/78 software package was anounced to go with the hardware. It is a subset of the next release of OS/8 that has been customized to run explicitly on this configuration. You should be able to plug in the distribution floppy disk and run without any configuration. The package includes most of the standard OS/8 package (except for FORTRAN II and certain other components that were not clear - maybe BUILD for example) plus BASIC and FORTRAN IV with a support library that has been shrunk to include only the parts that will be useful for this configuration (i.e. no DOUBLE PRECISION or Lab support routines and they say no COMPLEX support routines although some questions were raised to the effect that COMPLEX should work on this configsration and should be included). It seemed clear that MACREL and its LINKer as well as TECO will not be provided in the package. The new version of the monitor contains special features for the sake of OS/78 such as SET commands that patch the monitor and various other places to support a vidio type console terminal better (i.e. rubout is done with backspace, space, backspace and the TTY handler has an option that will pause for a couple of seconds after each 24 lines are output to let you look at it as well as the regular control-S and control-Q for start and stop). These modes are configured into the distributed version of OS/78 automatically but they will also be available in the regular OS/8 V3D release. In fact you should be able to run the regular OS/8 system on the '78 but if you need anything beyond what OS/78 contains, DEC seems to think you really want to have another, larger system to support the work and you will buy standard OS/8 for it.

The DECstation 78 package as shown contained the VT78 Processor/Terminal with 16K of memory, a dual diskette, a small cart that mounts both (plus the second dual drive if you get it) and which will roll through doorways easily, plus OS/78. This package is priced at about $8245 with Field Service priced at $68/month. An OEM version without the cart or OS/78 was priced at $7895 for single units. The cart is $250, the LA78 was $3300, OS/78 is $1400 alone but only $100 with a system, and the standard prices for RTS8 ($550), DECnet ($900), and MACREL ($350) apply. There is no Field Installation. The system is designed to be installed by the end user with no "special tools" in "less than an hour".

You should note that the extended memory control does not include the instruction trap so it will not be possible to run RTS8 with the OS/8 background task on this machine. Also note that the '78 has no "bus" as such at all. It is a self contained package. The five ports are all the interfacing that is possible and no memory expansion is possible. We were given some very interesting but brief information on configurations for word processing applications. In those packages you get the WPS/78 software rather than OS/78. If you wanted both OS/78 and WPS/78 software for one system, DEC said you would have to pay the full price for one of them (i.e. $1400 in the case of adding OS/78 to a WPS system). The details of what was being offered and for how much were too complex to try to report here without more information from DEC. The VT78 alone was not 'announced' or priced but it seemed likely that it will eventually be offered separately. The prospects for buying the CPU board alone were less hopeful. At this time it seems that DEC really does not like the idea of offering it alone. Maybe demand will change their minds in the future.

Several knowledgable users were heard to comment that the '78 seemed overpriced in today's market (some said about $2000) and it is arriving a little late but they still thought it was a very interesting and attractive offering. Some thought DEC was coming in high on the price to control what could be a large demand that might be hard to supply in the early months of production. I heard it said that the product development cycle for the '78 set some sort of a record for DEC once the Product Line finally decided it wanted to do something with the IM6100.

KT8A - 128K MEMORY EXPANSION CONTROL

DEC gave a session on their new KT8A extended memory control at the Spring Symposium. I requested details on the product from Jim Willis (manager of the PDP-8 Product Line) on the 16th of May so I could give complete and accurate details in the Newsletter but as of the 6th of July Jim has not even acknowledged my request. Therefore the following is only from my very limited notes.

The KT8A is a direct replacement for the KM8e extended memory control and instruction trap board. It is a hex wide board however that needs the full hex width connectors (it uses some wires in the extra two connectors to communicate the new extended memory addressing to new memory boards that can use it). For this reason it is mostly useful in PDP-8A boxes only. If you do not enable the new modes it should be just about perfectly compatible with the older extended memory controls. When the new modes are enabled it is possible to address 64K or 128K of memory. A hardware relocation of DMA transfers on 32K boundries is available (i.e. the FPP-8A could be run relocated in 32-64K, 64-96K, or 96-128K). This permits DMA transfers to the entire extended memory with no change to existing OMNIBUS interfaces. The relocation is separate for each DMA device (the priority wires are used to identify which one is making the transfer and which two bit relocation register to use). This feature also makes it possible to write an OS/8 background task for RTS8 or some other real-time or time-sharing system that could run FORTRAN IV with the FPP8A. In general there is a relocation feature that automatically will relocate (field-wise) a user mode program (i.e. "base and bounds relocation") through hardware implemented mapping of CDF, CIF, RDF, and RIF. With this feature in use OS/8 could run under a modified version of the RTS8 support task at an estimated 95% of full speed rather than the current 10% to 50%. (DEC did not say they were going to write either of these modifications to OS8SUP by the way.) You can choose to have the hardware map the first few fields (you have to start with field 0 I think) and trap references to higher fields if you like. You can choose to enable or disable trapping of other IOT instructions while you are using the memory mapping features. Unfortunately you cannot selectively chose which IOTs to trap as in some previous work in this area.

If you choose to expand to 64K you only give up device code 17 (i.e. IOTs 6170 thru 6177). All the new memory addressing instructions are squeezed into the unused IOTs in the existing 6200 thru 6277 series of extended memory control instructions. Actually the 4th or most significant bit of the field designation in 64K mode is just placed in bit position 9 of a regular extended memory instruction so, for example, CDF to fields 0 thru 7 is 6201 thru 6271 and CDF to fields 8 thru 15 is 6205 thru 6275. When you enable the extended addressing modes the KT8A starts responding to IOT 617n for various control and maintenance functions but if you stay in standard 32K mode device code 17 remains available for other uses.

When you expand beyond 64K you have to give up device codes 30 thru 37. They are used to address the upper 64K in exactly the same way as codes 20 thru 27 were used to address the lower 64K. Thus the 5th bit of the memory field designation appears in bit 5. The CDFs to fields 16 thru 23 are 6301 thru 6371 and the CDFs to fields 24 thru 31 are 6305 thru 6375 for example.

So we have up to 32 fields that are treated almost exactly the same way the old 8 fields were. It is now possible to address up to 128K in one program module with almost no increase in complexity over the old 32K versions. Compare this with the PDP-11 where extending a single program module to address beyond 32K is a very painful process that is rarely done even though it is badly needed for many applications.

For further details and prices you will have to contact for your local salesman.

SPRING SYMPOSIUM "WRAP UP-SESSION"

In the past couple of years it has become customary to hold a "wrap-up session" on the last day of the Symposium where the DECUS leadership (i.e. SIG chairman, DECUS officers and staff, and Symposium Committee) meet with DEC management representatives to review what has happened during the Symposium and to discuss each other's the concerns and needs. This aspect of user-DEC communication has grown to be viewed as a very important component of the DECUS-DEC relationship.

I regret to report that DEC's representitive to the the wrap-up session for the PDP-8 Product Line decided that she would not attend. After the session I spoke to her and she indicated that her actions were intentional and that she was aware of what she was doing. I feel that this was an insult to the 12 Bit SIG, to the Symposium Committee, and to all of DECUS. I hope that this young woman was acting on her own and not with the concurrance of the Product Line. Cooperation between the 12 Bit SIG and the various parts of DEC has always been open, honest, cooperative and mutually beneficial. It would be unfortunate and very disappointing to see that relationship replaced by a hostile, adversary one now. Let's all hope that the Product Line's reluctance to cooperate in recent months has been oversight due to preoccupation with the new developments reported elsewere in this issue and that they will now have time to restore the old, open, cooperative attitude that has always prevailed in the past.

I invite the PDP-8 Product Line to respond with their own view of this for the next Newsletter so we can all understand where they stand and what we can expect from our relationship in the future.

DECUS/US LIBRARY COMMITTEE

During the Spring Symposium the Library committee discussed offering standard packages of programs, custom packing services, media conversion services, improved catalogs and catalog costs, the impact of hobbyists on the Library, another Library price increase, user supplied media policies, requiring machine readable documentation for all submissions and progress of the evaluation of the TPL PDP-12 for possible acquisition by the Library.

Chuck Conley (DECUS Library Director) reported that in the six months since it was decided at the Fall Symposium to evaluate the PDP-12 the machine had never worked correctly and it has been impossible to get it repaired. Chuck favored dropping consideration of the 12 and perferred consideration of a PDP-11 for the Library. Some of the committee indicated that they could not accept the idea that it was impossible to get the 12 running and keep it that way and that they thought the machine had not had a fair evaluation as yet. It was decided to have Chuck and the rest of the DECUS staff give the evaluation one more try. As this is being written I understand a team from the Traditional Product Line and DEC's "In-house Field Service" are finally making a concerted effort to get the 12 up and running solidly.

If the 12 can be made to run reliably it could provide a full range of media conversion services, standard package creation and even custom packing services for the entire 12 bit community. It would let DECUS do a large part of its media copying for the other machines in-house and it would support some of the inter-library transfers that are desired (a FORTRAN program from the PDP-10 Library copied over to an OS/8 tape or floppy for example).

The price increase was tabled until the DECUS staff can come up with much better cost information for the Library and the rest of DECUS. It looks as though users will no longer be able to supply their own media when ordering programs as they have in the past. The reason this was done in the past was to keep costs to the user down but it creates many problems for the operation of the library (like getting bad tapes with orders and reduced income). The committee favored a review of pricing policies to go with this move however, as it was thought that the higher cost for DECUS supplied media was going to cut down on who could order and how much they could get.

Some effort is being made to find ways to improve the catalogs. For example we are experimenting with a combined KWIC index of all higher level language programs in all the libraries and various ideas for cross reference indexes of the individual catalogs. The cost of the 12 bit library catalogs is becoming a very serious issue. About 8000 copies of each half of the catalog and each supplement are sent out now. Printing and mailing costs are a major budget item. At present names are only deleted from the catalog list if mail to them is returned by the Post Office so the list just keeps getting bigger and bigger. Chuck's latest proposal on this is to mail the next catalog update to all the names on the 12 Bit SIG (i.e. this Newsletter) mailing list and to send cards to everyone else on the catalog list asking them to confirm that they still want to receive the 12 Bit Library Catalogs. Those that respond will receive the update. This proposal might change before it is finalized but the idea that the SIG members are considered to be sufficiently above the average level of interest to be considered for special treatment is interesting.

The IM6100 based imitations of the PDP-8 and the rumors about the posibility that Heathkit will offer a PDP-11 compatible computer kit stired up interest in the subject of how the Library should deal with hobbyist activities. No decisions were made but it is apparent that the committee will have to keep a close watch on developments in this area.

A little progress was made on the long standing problem of how the Library should deal with user generated modifications of DEC copywrited software was made. The problem stems from complex and poorly documented concerns of certain areas within DEC such as some product lines and the legal department with regard to maintaining the corporation's interests in the area of rights to their software products.

There was considerable sentiment in favor of some sort of requirement for machine readable documentation for every Library submission. I felt that on the larger systems where magtape, DECtape or other large capacity media are the rule and they are the standard form for submissions and distribution this idea would work better than on small systems where submissions may be on paper tape or other limited capacity media. Generation of machine readable documentation is made much easier with a program such as RUNOFF that will do the formatting of the finished document. The committee favored making RUNOFF the standard form for submission of such documentation since a more or less compatible version is available on the 16 bit and 36 bit systems. I pointed out that the only good, compatible RUNOFF for the 12 bit world was Tom McIntyre's (which I use for this Newsletter for example) but it is not available from DEC or DECUS. Tom sells it for a high enough price that not everyone will be able to get it just for DECUS Library documentation. If you have any thoughts on this subject or any of the others send me a note and send a copy to Chuck Conley at DECUS in Maynard marked for distribution to the rest of the Library committee.

DECUS/US PUBLICATIONS COMMITTEE

The Publications Committee met during the Spring Symposium. The principle topics of discussion were the various proposals for imposing charges for newsletters. It was reported that various people have been circulating proposals for a variety of schemes to this end. It seems that only a select inner circle has been in on most of this however, At the meeting the opinion was expressed that this was a subject of great importance to the SIGs and all of DECUS and that it should not be handled in secret. Several SIG representatives agreed that many of the proposals (some of which we still only know about second or third hand) would have immense impact on them and might even mean the end of their SIGs or at the very least, the end of their Newsletters as we now know them.

A long, loud, sometimes bitter discussion ensued during which it became apparent that no one in DECUS from the Executive Director on down had any real idea of what the true costs for any part of DECUS were. This was so in spite of a drive to determine that information that was initiated back while I was on the Executive Board more than a year and a half ago. No one is currently able to present a convincing case for any change or increase in DECUS charges. In the light of these discussions, the committee decided to take no action with respect to any recommendation to the US Executive Board on the subject of introducing charges for Newsletters (either in the form of direct charges or in the form of membership fees for SIGs or DECUS as a whole).

The need to regularly update the mailing lists for each SIG was discussed. As with the Library catalogs, no one is ever intentionally removed from a SIG list unless he so requests or the Post Office returns mail addressed to him. There was considerable agreement that a routine confirmation of your desire to continue receiving newsletters (perhaps once a year) would save money. Some question was raised as to whether DECUS could administer such a program however. It was not clear that the current mailing list system has has any provision for a way to remove a person's name from individual mailing lists (your name is on a master list once with flags for each thing you get like the various Library Catalogs, DECUScope, and the SIG newsletters you have requested). Isn't that a great way for a computer users society to have its mailing list system designed?

Another proposal to save money was suggested. Dr. Mark Lewis, the outgoing US Publications Coordinator, feels that the "new" DECUScope is not worth what it costs. (It is produced by paid DECUS staff rather than volunters as in the case of the newsletters and it is mailed to the entire DECUS membership rather than the limited membership of each SIG.) Mark would like to see DECUScope discontinued with the money saved being applied to the costs of the newsletters. This is a controversial subject and no real action on the proposal is likely for some time.

A related subject that was acted upon was the question of charges for back issues of newsletters. When new members join a SIG they sometimes request back issues to catch up on what has been going on or to get some article that they need that was published before they joined. Since the very first SIG was started (i.e. the PS/8 Special Interest Group that we now call the 12 Bit SIG) the back issue policy was for the DECUS office to always try their best to accommodate such requests without a charge. Last year the ILC (The DECUS International Liaison Committee - made up of one representative of each Executive Board) decided to institute a $5 charge for EACH back issue requested. This action was taken without consulting the US Publications Committee and I am not sure the US Executive Board or Publications Chairman were consulted either. (I do not think the DECUS bylaws anticipated that the ILC would take on this power - it was established as a "liaison" committee only - without any power of its own - but the various Executive Boards allowed this action to go unchallenged.)

Last Fall the Publications Committee voted to request that this absured back issue charge be rescinded, the justification for such a charge be developed, then a proper procedure under the bylaws be followed in determining what if any charge should be adopted. We learned at the Publications Committee meeting that that request has been ignored. The committee seemed very dissatisfied with that so as a result they voted to renew and upgrade their "request" to a "demand".

It has been noted that just about all the newsletters are published under the auspices of the US Chapter so the US Publication policies are the de-facto policies for all of DECUS. In effect the US Board and its Publications Coordinator could, if they choose, unilaterally set newsletter policies and procedures as a result. There was sentiment expressed at the committee meeting to the effect that if the ILC continues on its present, unresponsive course in these matters, the US Chapter should consider such unilateral action to deal with the resulting problems.

If you have any thoughts on Publications issues you should send me a note and also send a copy to Ely Glazer, DECUS Executive Director, in Maynard, marked for distribution to the US Publications Coordinator and Committee. If you are in one of the other chapters you will also want to request distribution to your own Publications Committee and/or Executive Board.

FOCAL SIG

I attended an organizational meeting of the FOCAL SIG. There was a suprisingly large turnout with representitives of many different versions of FOCAL on almost all DEC computers and operating systems as well as some others. I hope the organizers will keep us informed of what is happening because the 12 Bit family is where FOCAL started and a large part of its support comes from our ranks.

RECENT SUBMISSIONS TO THE DECUS PROGRAM LIBRARY

Jackie Page, the DECUS librarian for 12 bit programs, has sent me the abstracts for the following recent submissions.

The following item is from Ronald P. Larkin at Rockefeller University. It is for OS/8 and a Tektronix 4006-1 Graphic Display Terminal on a serial interface.

GT.PA - OS/8 Handler for Tektronix 4006-1 Graphic Terminal as Console Device (DECUS 8-866)
"GT.PA is an OS/8 handler for the Tektronix 4006-1 Graphic Display Terminal in alphanumeric mode. It allows the terminal to input and output as the console device (device codes 3 and 4), possibly replacing a teletype in this capacity. The standard OS/8 features are available, plus the added feature of stopping at the bottom of the screen during output, allowing the operator to hit any key in order to erase and refill the screen with the needed section of the text. The handler was created by modifying the DEC KL8E.PA handler. It occupies two pages."

The following two items are from David Spector at DEC in Maynard. They are short routines written in PAL that will work on any PDP-8, PDP-12, VT-78, or similar CPU.

RANDOM.PA - Random Number Generator (DECUS 8-867)
"This stand-alone subroutine generates a well-distributed sequence of pseudo-random words. It is very fast (an average of 13 decimal instruction executions per call)."

MLDV - Multiplication and Division Subroutines (DECUS 8-868)
"These four subroutines do the following single-precision, unsigned operations:

1. Integer Multiplication
2. Fractional Multiplication
3. Integer Division (with Remainder)
4. Fractional Division

These are suitable for any PDP-8 family computer, including the VT-78. Full descriptive comments appear in the source."

The following item is from P. H. Holtham at the National Research Council of Canada in Ottawa. It is a one page OS/8 device handler that uses a 7 track Magnetic Tape Drive.

PDP-8 to IBM Transfer via 7-track Magnetic Tape (DECUS 8-868)
"An OS/8 handler for writing ASCII files or output onto 7-track tape is provided. Character unpacking and tape blocking are done within the handler. A further program for reading the tape into, for example, an IBM computer, is also given. Both programs have the capability of handling variable record length files."

The following item is by Ron Lesser, Gary Schick, and Len Zablow from the College of Physicians and Surgeons at Columbia University in New York. I understand it is the first LINC-8 program submitted since 1973. It runs on a 4k LINC-8, is written in LAP6W and requires DECUS L-40.

AVE4+ - Four Channel Averager Plus Triggered Simulation (DECUS L-126)
"AVE4+ is a modification of the average 2 portion of AVERAGE, DECUS

program L-40, written by Dennis J. Nichols. In addition to averaging four channels of data, it has provisions for timing the initiation of the averaging sweep and for timing the delivery of stimuli to the subject. Each can be delayed in reference to a threshold crossing of electrical actvity from the subject."

The following item is from Loe Bowbeer at Clinton High School in Iowa. It runs under FOCAL 69.

XROOTY - Xth Root of Y (DECUS FOCAL8-337)
"The program XROOTY (Xth root of Y) uses an algorithm based on an iterative process to calculate the integer roots of numbers."

The following two items are by Christopher A. Kryzan at Northwestern University. They run under EDU-30 BASIC in 4k. The first item also requires a DF-32 disk.

EDITOR - Symbolic Editor Program (DECUS BASIC8-92)
"Text-editing and word processing facilities are welcome and desired on all computer systems, including small systems with only one available compiler at one time period. In order to provide editing capabilities on even these small systems, EDITOR was created. BASIC was seen as one of the most abundant system languages in use on small high school systems, and thus EDITOR was designed in the BASIC language. Text-editing capabilities similar to standard DEC editors and a character capacity of up to 6600 characters serve to enchance EDITOR's attractiveness."

SCRMBL - Scrambled Word Generator (DECUS BASIC8-94)
"Oftentimes instructors wish to supplement their lectures with extraordinary teaching aids. One common method utilized by teachers is scrambled word lists. In order to increase the ease with which lists can be compiled, SCRMBL was created. This program will scramble words in lists of up to 150 characters (or more on larger computer systems). An attractive feature of this program is its ability to generate multiple copies for mass distribution."

The following item is by Ron Voit at Northweastern. It runs under EDU-30 BASIC in 4k and would like a line printer but does not require it.

MADMAZ - Maze Generator (DECUS BASIC8-95)
"Computers have many non-scientific applications in addition to their technical side, one of which is found in demonstrations and gaming. An interesting sub-genre of this is the construction and solution of puzzles. MADMAZ is designed to create 15 x 15 maze puzzles, replete with solutions as well. Execution can be quite lengthy, but the results are well worth the wait."

The following item is by David C. Nicosia at Northwestern. It has the same requirements as the previous item.

REDPUN - Paper Tape Message Generator (DECUS BASIC8-96)
"A variety of programs to produce punched paper tape messages have been

published, but this particular version proves to be one of the most efficent yet designed in BASIC. The program consists simply of a data list of characters and a routine to enter and output the message, creating punched tape records of up to 400 characters in length."

The following item is from Pascal Chesnais at the United Nations School in New York. It runs under EDU-25 BASIC and requires a VT-05.

CLOCK - Simulation of a digital clock on the VT-05
"This program will simulate a digital clock on the VT-05 terminal using the x,y plotting grid already programmed into the VT-05."

The following item is from Jerry N. Rabinowtiz at the Claymont School District in Delaware. It is in "BASIC" and requires 8k on a "PDP-8/I".

BOWL1 and BOWL2 - Bowling Record Tabulator (DECUS BASIC 8-100)
"This two-part program will tabulate weekly records for a bowling league with twelve four man teams; but, can be used for leagues with any number of teams, and any number of bowlers. It will run under virtually any version of BASIC -- NO string handling capabilities are required."

The following item is from Brad Tebow at the Camelback High School in Phoenix. It is written for BASIC on the H-P Series 2000F computers!

BATNUM - Battle of Numbers
"Battle of numbers uses a very simple formula. It can be beaten if the player knows the formula or is very lucky."

The following items are from Joe Bowbeer at Clinton High School. They are in "BASIC" (CHANGE specifies EDUsystem-50) and are all cataloged under DECUS BASIC8-102.

EPSQRT - Extended Precision Square Root
"The program EPSQRT is approximately twice as precise as the BASIC "SQR" function; the precision averages at about fifteen significant figures. The user specifies the total number of digits to be printed in the answer."

EPLG10 - Extended Precision Log Base 10
The program EPLG10 is a short but very effective routine to determine the common logarithms of numbers. The logs are printed to approximately ten significant figures. The user specifies the number of digits to be printed in the mantissa."

POETRY
The program POETRY composes free verse. It uses knowledge of sentence patterns and randomly selects words to assemble them in lines, as opposed to randomly picking whole or partial phrases. The poems follow the form A-B-A-B, and because of the words chosen they all portray electrical machines taking over the world and quarreling among themselves."

GRAPH0,GRAPH1,GRAPH2 - Ellipse and Circle Plotting
"The programs GRAPH0,GRAPH1, and GRAPH2 plot circles and ellipses of

desired width and height in only one pass per line (to save the most time). GRAPHO displays the desired form on an x- y- coordinate system best suited for plotting integer values. The square of each x-coordinate is also given. GRAPH1 plots only the outline of the ellipse or circle, omitting the coordinates and numbers. GRAPH2 is identical to GRAPH1, but solid forms are plotted. GRAPHO plots nearly perfect ellipses and circles, while the graphs of the latter two programs, not hindered by the coordinates, are perfect.

CHANGE
The program CHANGE uses the Edusystem-50 CHANGE statement to write an inputted line of up to forty-eight characters backwards. This routine may prove useful incorporated into many other novelty and game programs."

SEQUES,GEMSQU - Arithmetic Sequences, Geometric Sequences
"These two short programs solve for unknown values in two types of sequences. The program SEQUES finds unknown terms and the common difference in an arithmetic sequence where two terms are known. The program GEMSQU solves for any unknown terms in a geometric sequence if two terms are known. The common ratio is also found."

COEFNT,PROBTY,ESPTST - Coefficients, Probabilities, ESP Test
"The program COEFNT gives the coefficents and exponents for a binomial expansion of desired degree. Degrees of one hundred or more can be solved for because the factorial algorithm is not used."

REPEAT - Repeating Decimals
"The program REPEAT finds the lowest terms fraction determined by a repeating decimal. Any combination of the following may be entered: (1) integers, (2) non-repeating decimal leaders, (3) repeating decimals"

MEDIA CONVERSION OFFER

Paul F. Sullivan has written to offer help to those who need media conversion services (i.e. you need something copied from DECtape to RK05 disk, etc.). He has available paper tape reader, RX01 floppy disk, DECtape, and an RK05 disk. He will supply intermedia conversions for the DEC price for the media plus a nominal service fee. For information contact Ann H. Silva, C20, Acushnet Company, P.O. Box E916, New Bedford, MA 02742.

FROM BRIAN CONVERSE

COMMENTS ON THE RECENT DISCUSSION OF U/W FOCAL, BASIC AND FORTRAN

In the beginning, I suspect, most PDP-8 users were a bit awed by the computer power they had purchased and worried little that everything occurred on paper tape or that each program was on a separate one. DEC was selling hardware and little else at the PDP-8 level because it worked- they made money. Later, they began to offer such items as FOCAL and FORTRAN-II as enticements. The user, and more importantly, purchaser, of a minicomputer system was more often impressed by hardware than software and tended to write most of his early systems in assembler with little resistance to such a task. The selection of a minicomputer was and often still is in the hands of a single user. After all, this thing will often not do any rigorously defined task, but turns out to do something that in many cases neither the purchaser or anybody else ever thought of. Most "big" computers (whatever that means now) are selected via the most mind-boggling array of meetings, evaluation criteria, budget requirements, outside consultants and outright bribery imaginable- but these are easy to pin down: They print bank account statements or mix breakfast cereal or something. It is in some respects unfortunate that this is so, since many mini purchasers are more impressed by hardware features than software features, at least before the mini arrives and often, in defense before others, afterwards.

Some users, at universities perhaps, became acquainted with software for large machines through one means or another. Perhaps the most bothersome was when a starving computer sciences student was hired to shepherd the various manifestations of paper tape FORTRAN through the low-speed reader and with disdain murmured about 'disk operating systems' and OS/360 or EXEC8 or META4 or something. With impetus from users DEC began its long adventures with minicomputer operating systems. By this time, the fever was rife; the 'ante" for a minicomputer was by then not only programming languages but this "operating system" thing. By the mid-70's not only is the hardware terribly powerful, but you simply can't start or keep afloat a mini company without something akin to OS/360 available (at a stiff price) for the top of the line. There is even a form of RSX-11 for the LSI-11! Depending on your point of view, this is humorous, tragic, or very innovative -with the later view predominating.

The FPP-12 is variously, to me, at least, a godsend, an albatross, a boat anchor, and a hideous mistake. The major hassle with the FPP-12 at this location is that it is very difficult to get PDP-12 ADC values into and out of it; the fix and float functions ain't so hot. A 32k PDP-8 (we have only 16k) is quickly core-bound when you start averaging signals and each value uses 3 12-bit words... Double-precision floating point is admittedly nice, especially its complex number capability, but now the requirement is for six 12-bit words per value! Learning FPP-12 language is admittedly not all that hard, but our PDP-12 runs three languages for LINC mode, PDP-8 mode and FPP-12 mode. There are assemblers which can handle two modes at once (PAL-12, RALF) and there are optimized assemblers for a single mode (PAL-8, FLAP), but nothing with a consistant and tolerable syntax which handles all three. I/O, as I mentioned previously, is such a pain via the FPP-12 that some PDP-8 code is usually required and adding it in RALF is made so much trouble you'd think you were paying off some heinous crime committed in a past life. There have been several letters about programming the FPP-12 in recent newsletters (18, 21) of the order, 'Watch out or this will get you!'. This is not what an assembler is all about. Perhaps something like RALF could be made more intelligent to the point that it lists possible (not definite) trouble points. PAL-8 does it in a primitive way, for example. It would additionally be nice if up-to-date sources were available unbundled from the rest of OS/8 and loose enough in coding so that they could be extended by users (to include LINC mode, for example). (Note: with luck, MACREL will allow you to write macros to do most of this. I think many of the RALF problems come from the very poor documentation and FORTRAN IV's structure and calling conventions rather than from the assembler itself. R.H.) Some wish to "upgrade" to the PDP-11, but it is not always possible to get a programmable, independent FPP, and the boost to 16 bits is not really that impressive- try 24 or 32 bits, or plan on it since hardware costs are coming down rapidly. All you get with the 11 is the possibility (if DEC dosen't stop tinkering with it) of a de facto industry standard architecture and instruction set.

The major factor at most locations is not which language is best, but which language is most popular since you have access to a bunch of experienced users, necessarily up-to-date documentation, and the latest version of the language the site can afford. The next consideration is "can I take it with me?" This question as applied to FOCAL and BASIC is a very important one; less so with FORTRAN. FORTRAN IV is definitely a bitter pill to swallow when ordering a PDP-8- you gotta pay extra not only for the software (let's hear it for those SPR's that say "arctangent is bad, but all you have to do is patch the source code [which you undoubtedly have shelled out many $$$ for right?] and reassemble [and sit through another session with LIBRA to insert] and it is again OK") but also for more core so you can run it once it arrives (preferably lots more core unless you want to learn about overlays real quick). (Note: I don't agree that FORTRAN IV uses that much more core for a given program than the other languages. In my case, I do find that it is much easier to write big programs, so I end up using more core for that reason. Most of the available languages use about 4k for the run time support, all use at least F IV's 3 words of storage per variable (except for F II's 12-bit integers), and the stored program code is not all that much more or less core efficent in the various choices. On the other hand, F IV does offer the chance to use overlays if you have to have them! R.H.) But some who complain about the cost of adding FORTRAN IV easerly defend the need for a mass storage device and 8k before OS/8 will fly. FOCAL is a fine language, but it is hard to find elsewhere. BASIC is easy to find, but never looks the same in two places. Even FORTRAN is always manifesting itself in different enough forms to cause major conversion problems. Each manufacturer adds "superset" features that you suddenly discover aren't there on someone else's software. (Note: OS/8 FORTRAN IV is one of the best in this regard, i.e. it has very few "superset" features that will not be available elsewhere - I am not sure if that is good or bad! R.H.) The basic argument is that it's very hard to make a case for any high-level language over another across a broad class of potential uses. However, unlike some aspects of life, people are not so committed to one particular language to the point of violence, so a discussion of the comparative virtues and faults is usually entertaining.

Regarding the concern in various issues over what DEC will and will not do about PDP-8 software, I suspect that the return on a major DEC PDP-8 software effort will not be as high as if they committed the resources to PDP-11 work because they'd likely sell more 11's on the basis of the new software than 8s. As Bob Dylan is reputed to have said, "Just because you like my stuff dosen't mean I owe you anything", and the fact there is a huge installed base of PDP-8s that would buy new software can't beat the fact that there is also a pretty good-sized base of 11s and potential buyers that would be lured by the new software. Given an unlimited amount of capital, DEC would likely follow both courses, but in a crunch do you think they'd opt for more 8 software? The outside world can often outdo DEC at software: U/W FOCAL is a case in point: the trick is to achieve some coordination, perhaps thru DECUS. DECUS members often have very low-cost programming resources (slave labor) and there are many PDP-8 users of which a few must be hotshot compiler/interpeter/OS designers who could at least make up the skeleton. Since modular programming is "in" these days and PDP-8 programmers claim renown for their "tight" coding, groups or single programmers could take on responsibility for coding small segments of the new software which could be tested, verified, and so on until the product was finished. Then we'd have software that was easy to work on, was up-to-date, would be inexpensive with available sources, and would fit into a minimal amount of core.

The letter in Newsletter 22 about GR test systems using PDP-8s and OS/8 was very interesting. So many computers now go out into the world bolted into the underside of a framis ejector or orbiting debris locator and do only one task- if users realized that there was OS/8 sitting there to do other things maybe they'd do them when the intended task is idle or disconnected. The note about the XY plotter interface was interesting as we suddenly began to have difficulty with our Houston "OMNIGRAPHIC" plotter in March of 1976. This was not a DEC peripheral, so there was some suspicion that the mechanical part of the Houston couldn't hack it. Finally, a 'scope revealed that the PLPL and PLPR were pulses about 5 microseconds wide; the Houston wouldn't swallow PLPL. Also, using a function generator, we were able to run the Houston offline at more than 500 steps per second (with square waves @50% duty cycle). (Note: Many of the DEC plotter interfaces are actually set up to run at 200 steps per second rather than 300 which is the nominal capability of a Calcomp 565. On my interface the timing would not be hard to change - just a timing capacitor on a one shot. R.H.) The culprit was software: We had hooked up the Automatic Priority Interupt into our PDP-12 plot program which fed plots to the plotter interface as fast as it would take them. The electronics in the Houston did pretty well, considering, but after a few years, age and parts tolerances led to one or the other of the pen drivers being unable to react to such very short pulses. Once the patch was installed, we were able to "turn up" the delay before interrupt (our plotter, and, I suspect, the 565 as well, does not signal "movement done" so the interface waits an appropriate time) and run the plotter much faster than it had ever done before.

Brian Converse
EPA Clinical Studies Division
Environmental Protection Agency
Mail Drop 58
Research Triangle Park, N.C. 27711

NOTE FROM STUART DOLE

Stuart wrote with a few thoughts and suggestions. He says that although

he has submitted some patches and enhancements for OS/8 BASIC, he really does not like it (1) because it is very difficult to structure programs, although the ability to leave off line numbers and use tabs and form feeds in EDIT or TECO prepared programs helps, and (2) because it is not interrupt driven and all his applications are essentially real-time. What he would really like to do is use FORTRAN IV but there are a number of serious problems. First, he needs to be able to dynamically open and close files as well as test for their existence and size, and second, he is concerned that since F4 doesn't do any true integer arithmetic it may not be any faster than BASIC for "number crunching". (Note that Bob Phelps' USR solves most of the files problems and is available from DECUS but this is no excuse for DEC not doing the job right - RH).

Stuart says that he feels a very real need for the 8 is a good high level language, where the basic constructs of structured programming would be primitive elements, and where there would be manipulation of extended data types. He says PASCAL comes to mind as a good canidate. To really do it right, he thinks there should be the usual 3-word reals (floating point), 12-bit integers, boolean, character, and string variables, as well as pointer types, device regester address types, and files (of various sorts). One should be able to lookup, enter, and close files easily. If a file is not there it should not be a fatal error, and it should be easy to set file size. File I/O should be possible in one of several modes: character, real, and 12-bit. The latter two ought to have direct access capabilities. Stuart thinks the language "C" on the PDP-11 might be a good model. He considered Roger Abbott's ALGOL, but the limited file and string capabilities ruled it out even though its speed was very nice. (I would like to add to the list my desire to be able to trap all error conditions within a program - even control-c - so that an application that is written for non-computer types to use will not have to have complex instructions for error recovery. For example, every time you type an alphabetic character while FORTRAN is trying to do numeric input you get an error that exits to the monitor with out saving your files. Your program never gets a chance to close and save the files, give the operator a second chance or a prompt, try to correct the error, or any of the other possible actions. R.H.)

8K BATCH

Jim VanZee recently sent along a copy of a write-up on Eduardo J Subelman's BATCH8. It is a program designed to be very compatible with most of the features of DEC's BATCH for OS/8 but it runs in the standard minimum 8k system rather than requiring 12k of memory the way DEC's BATCH does. Jim says that Mr. Subelman is not interested in "giveaways" but at this point we do not know anything further about his plans for BATCH8. The write-up is a brief but fairly clean and complete addendum to the standard DEC documentation for BATCH which explains how BATCH8 works, how to get it going, and the diffrences between it and DEC BATCH. Jim indicates that his first experiences with BATCH8 have been favorable. Mr. Subelman is at The University of California, Berkeley, College of Enginearing, Dept. of Industrial Engineering and Operations Research.

LONGER DECTAPES

(It looks as though the interest in squeezing more and more data onto a LINCtape is spreading to DECtape. For example I recently received the following letter from Michael E. Mazzoni.)

"How long is a DECtape?

The "Small Computer Handbook - 1973" says a PDP-8 DECtape is formatted for 1474 blocks of 129-12 bit words per block. Ahead of and behind each data block is an interblock zone of 5-18 bit words (10 between each data block). The smallest division of a DECtape is a line, or 3 bits. Thus each data word uses 4 lines and each word in the interblock zone uses 6 lines. The total number of lines for PDP-8 format is:

```
          1474 * 129 * 4 = 760584
        + 1474 *  10 * 6 =  88440
                TOTAL       849024 (Small Computer Handbook says 849036)
```

The "PDP-11 Peripherals Handbook - 1973-74" says a PDP-11 DECtape is formatted for 578 blocks of 256-18 bit words per block, with the same number of words in the interblock zone. The total number of lines for PDP-11 format is:

```
           578 * 256 * 6 = 887808
        +  578 *  10 * 6 =  34680
                TOTAL       922488
```

The PDP-11 puts 8.7% more lines on the same DECtape. This confusion is perpetuated by DEC in the form of a double standard in the DECtape formatter (DTFRMT.SV). Who's right? Or more sensibly, why are there two "lengths" to the same physical piece of tape?

Optimistically, there are 922488 lines on a DECtape, then the following changes to DTFRMT.SV and PIP will allow PDP-8 DECtape to be formatted for 1600 blocks of 129-12 bit words per block, giving 800 OS/8 blocks per tape (less directory and system blocks).

```
        .GET SYS:DTFRMT
        .ODT
        1573/7161 7560  ;-201-17
        1574/1011 3077  ;-3100+1
        1000G
```

The response to "MARK" will now be "0201 WORDS,3100 BLOCKS.OK..."

Actually 1574 can be changed to any value. Chuck Lassner tells me that he regularly formats for 1728 blocks (loc 1574=3177).

A word of caution, though: DECtapes are no longer advertised as 260 feet long, and the tapes shipped me in the last year are definitely shorter than 260 feet. When I format PDP-11 tapes I have to use only one wrap of tape around the right hand spool when mounting the tape; even then most tapes run off the end during the first pass.

Change PIP as follows to zero a DECtape for 800 blocks:

```
        .GET SYS:PIP
        .ODT
        13616/6437 6340  ;-1440
        1300G
        *...etc...
```

I want to acknowledge that most of the technical information needed to write this letter was graciously supplied by Chuck Lassner."

(Mr. Mazzoni is president of Process Control Systems, Inc., 18130 S. Thornapple Lane, New Berlin, Wisconsin 53151. If you are interested in working with non-standard length tapes (or other storage media) under OS/8 you can avoid having to keep track of how long each one is and having to patch PIP to zero (and squish?) them by using PARAM, ZERO, and SQUASH from DECsystem/8 in the DECUS library. There is a special feature included in the design for just this purpose. Maybe you have old DECtapes that have the first few blocks (directory or system area) worn out but the rest of the tape is OK. Using these techniques you could reuse them by cutting them down and re-formating them with fewer blocks. R.H.)

"CHANGE" FUNCTION FOR BASIC

(Dr. O. Arthur Stiennon contributed the following subroutine for BASIC. His address is Park-Regent Medical Building, One South Park Street, Madison, Wisconsin 53715).

Neither OS/8 nor EDUCOMP (i.e. OMSI) BASIC have a "CHANGE" command to convert a string to the corresponding number. The 5-line subroutine embedded in the following demo will make this useful feature available. (Note: As written, this subroutine works for EDUCOMP BASIC, it needs a few modifications for OS/8 BASIC. See also the OS/8 BASIC "VAL" function. R.H.)

```
100     PRINT "ENTER A NUMERIC STRING";: INPUT A$
110     L=LEN(A$)
120     GOSUB 200
130     PRINT "THE CONVERSION IS  ";: PRINT S
140     PRINT " THE STRING IS  ";: PRINT A$: PRINT: PRINT
150     S=0
160     GOTO 100
200       REM *** 4-LINE CONVERSION SUBROUTINE ***
210       REM *** CONVERTS 'A$' TO THE INTEGER 'S' ***
220       REM *** CAN'T USE DECIMAL INPUT ***
230       FOR J=1 TO L
240         LET I=ASCII(MID(A$,J,1))-48
250         S=S+I*10^(L-J)
260         NEXT J
270         RETURN
300     END
```

VIURAM-8/12

Eric Wogsberg sent some information on an interesting video terminal interface for the OMNIBUS (he does not say if it will work on the DW8e bus adapter for the older machines). The basic product is a single quad board that plugs into the OMNIBUS and looks like 1k of memory which can be set to any addresses you choose. Connected to the memory is a video character generator that makes composite video that you can feed to a standard video monitor. The display gives 12 lines of 80 characters. A program can update the display by addressing the corresponding "memory" location. Keyboard interface and so on are also available. The software provided includes:

a) patches to TECO to display the text buffer
   before and after the cursor.

b) a screen editing program which is useful
   when experimenting with diffrent screen formats.

c) an OS/8 device handler which allows keyboard
   control of the output (start, stop, scroll rate,
   page or half-page, search for string).

This seems like an interesting, low cost possibility for certain applications. It is slightly reminiscent of the VT8e although it provides it's own refresh memory and needs separate display and keyboard but it does not cause any overhead due to Data Break access to the main CPU memory as in the case of the VT8e. For information contact Computer Technology, 6043 Lawton Ave., Oakland, CA 94618.

HELP

The following request for help recently made it's way to me.

"We have recently had a PDP8I donated to us as a school by a local electronics firm. Some of the boards are missing but the main frame, control panel, and power supply are in good shape. We are interested in approaching this from one of the following angles: 1) Picking up the necessary boards and peripherals to make it useful or 2) selling what is available and turning what we can get out of it into the beginnings of another system that we can use.

We would be interested in picking up any or all of the following literature:

a) Small Computer Handbook for PDP8I
b) Programming Handbook
c) Logic Handbook
d) OS/8 Operating System Manual

We are also interested in Peripheral and Interfaces for Teletype, Video display, a Card Reader, and Cassette.

What we are able to obtain will be the beginnings of a Computer Techincian Program.

G. Leon Pully
Instructor, Davis Vocational Center
45 East State
Farmington, Utah 84025"

PATCH TO THE PATCH FOR INITIAL COMMAND EXECUTION ON STARTUP FOR TD8E

The January issue contains a patch (p. 21) which allows a CCL command to be executed automatically upon system bootstraping. In order for this patch to work on systems using TD8E DECtapes as the system device a slight change must be made. The line which reads R,0 should read R,66. Otherwise make the patch just as indicated." (Submitted by Benjamin A. Fairbank from the Department of Psychology, Box 5095, Las Cruces, New Mexico 88003)

ERRATA

In the last newsletter (number 22) on page 7, the statement that produces the error was mis-typed. It should have been:

```
DIMENSION A(100),B(100)
EQUIVALENCE (A(100),B(1))
                    ----
```

Thanks to Doug Parmenter for pointing out the error.

FIX FOR INCREMENTAL PLOTTER INTERFACE PROBLEM

Dan Smith sent a copy of the information on how to deal with the plotter interface problem he reported. (Reference Tec Tip XY12-TT-1)

"The XY12 and XY8I both use the M704 module. If the IOT 6501 to check the plotter flag is given too soon after issuing a Pen Right or Pen Left instruction (6511 and 6521), the flops for these instructions are cleared and the plotter doesn't respond."

"A fix for this is to disable the gating on the clock inputs to the Pen Right and Pen Left flops when device code 50 is used. This change is to the M704. Cut the etch from: E0306 to E0309. Add 30-gauge wires: E1508 to E0309, E1503 to E1511, BH2 to E1510, E1804 to E1509."

RTS/8 WORKING GROUP NOTES FROM LEE NICHOLS

The "RTS/8 Workshop and Papers" session at the Boston Symposium was well attended and started off with two excellent application papers. Steve Root, of the PDP-8 Software Engineering Group, then described the new features planned for RTS/8 Version 3, which will probably be released this time next year. Steve is now the programmer responsible for the development and maintenance of the RTS/8 system. The remainder of the session was an active discussion of user ideas and problems in using RTS/8. The RTS/8 Working Group met with Steve that evening and proposed several additional features for Version 3.

For Version 3, the RTS/8 tasks will be restructured to assemble with MACREL. Major new modules will include a sysgen dialogue for system configuration, software to support the new KT8-A extended memory hardware, and a much faster TTY handler. Details of the sysgen procedure are sketchy so far, but basically, the program will ask a series of questions about the possible machine configurations and generate three files from the user responses. The files generated will be a parameter file, a batch file which will assemble the required task modules with MACREL, and a second batch file for the linking loader. There will also be some method of adding user tasks into the sysgen dialogue. Initially the KT8-A support code will probably provide a foreground area of up to 64K, and a single background task of up to 32K. Using the KT8-A, OS/8 background programs should run at about 95% of their stand-alone speed. The new TTY handler substantially reduces the executive overhead per character at a slight loss in interrupt response time. The current TTY handler begins to load down the system when driving a 9600 baud line, particularly when DECNET is in the system.

During the meeting of the RTS/8 Working Group, many good suggestions were discussed for enhancing the RTS/8 system. The use of MACREL will cause some compatibility problems, so now is the time to make any needed syntax changes. After the meeting, Steve tentatively agreed to incorporate the following changes in Version 3:

1) Add a parameter and the conditional code to save and restore the EAE step counter when switching context. The step counter will not be saved at interrupt time. (This will add 2 words to the state table for each task.) A separate parameter will be used to conditionally save the MQ register, when an EAE is not in the CPU.

2) Modify the CLOCK handler to: Remove a request from the clock queue when that request is canceled. Post a specified event flag in addition to the CLKMSG event flag after the requested interval. Add code at interrupt level to allow clock rates faster than 5 milliseconds, probably 1 millisecond (1KHz) maximum.

3) Modify MCR to: Log time in seconds, allow 6 characters for task names. Add code to list up to 127 tasks in the status list. (It turns out this is the only real code addition that is required to allow 127 tasks in a RTS/8 system. There is usually one mask in each task which must be changed from 0077 to 0177, but 7 bits were always allotted whenever task numbers are used in CAL's.)

4) Modify the TTY handler to: Add the code to share the TTY handler between foreground and background tasks. Also add some method for the foreground to break through for high priority messages. Maybe add some sort of timeout on keyboard requests.

5) Change the parameterfile for the SWAPPER task to include the names of the non-resident tasks and their partitions. The SWAPPER task would look up the block addresses of the non-resident tasks automatically during system startup, instead of having the user enter each non-resident task name to the command decoder.

6) Add the field of the text to message packets for non-file structured devices (TTY:, LPT:, etc.), or at least those packets which do not contain text, but point to a text string or buffer. This will be required for MACREL and will be an incompatibility with Version 2B. Currently these text areas must be in the same field as the task that sends them. Having the field as part of the message packet will allow passing packets from task to task and greatly simplify spooling output messages for non-file structured devices.

7) Add some safety checks for the CAL arguments to trap illegal task numbers and arguments, and non-existant tasks.

Other ideas which were discussed that might be beyond the scope of Version 3, but are good user projects are:

1) Modify the MCR to be a receive only task. This would free the TTY handler automatically and allow user tasks to use the MCR services. For example, a user task could ask the MCR to log the date and time, or request the time is ASCII to add to a message or data file. This will also require modification to the console terminal handler to collect unsolicited keyboard input and pass it to MCR.

2) Add conditional code to the MCR command decoder so that if MCR does not recognize a command entry, it would pass the entry to a user task. This would allow quick and simple expansion of the system commands without the user modifying MCR. This technique is used for the EXIT task in Version 2B.

3) Create a single KL8A handler which can control all 4 serial lines.

4) Split the current TTY handler into two tasks, one for the keyboard and the other for the printer. This would allow output to continue while the keyboard task waited (maybe forever) for user input.

5) Change the OS8SUP task to create and send text messages to the TTY handler instead of doing I/O directly. This might be the cleanest way to share the TTY handler between the foreground and background.

6) Change the OS8SUP/OS8F interlock to remember which devices OS/8 is using, and allow OS8F free access to any other devices in the system. We really need a new concept for this foreground/background interlock! Any suggestions?

7) Have RTS/8 update the OS/8 date work at midnight.

If you have any other ideas for enhancing RTS/8, drop me a note, and I'll see they get to Steve, and I'll publish them in future newsletters. Now is the time for new ideas while the specifications for Version 3 are being prepared.

Steve will make every effort to release Version 3 as bug free as possible, but to do this he needs to know about any bugs users may have encountered, and maybe fixed, without submitting a SPR. If you have found a bug, document it as a SPR and get it into the system. Also, please send a copy to me. If that is too much trouble, write me a note describing the bug or give me a call. My address and phone number are below.

During the RTS/8 Working Group meeting, the subject of half-bugs came up. A half-bug is one you think you have, but can't pin down to a particular task or sequence. If only one person sees a half-bug, it may be a user error or hardware problem. But if several people see the same half-bug, it becomes something to look into. If you think you have found a half-bug, drop me a note. If several users report the same half-bug, I'll pass it along to Steve.

A half-bug that was mentioned by several users during DECUS, seemed to indicate that the RTS/8 scheduler would occasionally not execute a task which became runnable as a result of an interrupt service routine request (POSTDS). The runnable task would not execute until some other event causedthe scheduler to run. Both Steve and I took a long look at this problem but could not make it happen. I replaced the NULL task with a task which repeatedly scanned the task flags table looking for zero (runnable) entries. After about 30 hours of running this test program in a busy system, no scheduler errors had been encountered. If anyone has seen a problem like this, or has any ideas about how to trap it, please contact me.

The next issue of Digital Software News will document some Version 2 bugs. I'll try to write up the unpublished SPR's for the next newsletter.

I would like to formulate some guidelines for submitting RTS/8 tasks to DECUS in order to make them easier to integrate into user systems. Any suggestions?? My initial thoughts are:

1) Use and document symbolic definitions for memory field and program origin.

2) Explain what modifications, if any, are necessary to other RTS/8 tasks.

3) Explain what effects the new task has on other tasks, if any.

4) Discuss suggested task priorities, and document the reasons if this task needs a special priority.

5) List any other tasks which the new task requires that are not supplied in the regular RTS/8 release.

I would like to compile a list of RTS/8 users who have the time and interest to assist in the working group efforts. We need people to field test software, edit new documentation, etc. If you are interested, send me you name and your computer configuration. My address is:

Lee Nichols
E. I. Du Pont
Experimental Station
Building 357
Wilmington, DE 19898
(302) 772-3839

LHN:jmm
6/23/77

FROM DAN SMITH
Eye Research Institute
20 Staniford St.
Boston, Mass. 02114   617 742-3140

A PATCH TO MAKE FORTRAN IV LIBRARY DEFAULT DEVICE OTHER THAN SYS:

The Fortran IV loader assumes that the library is SYS:FORLIB.RL unless a different library is explicitly specified via the /L option. It is often convenient to have the library reside on a device other than SYS:. For example, on a DECtape-only system, significant time can be saved by having the library on a unit other than the system (since this shortens the distance between the compiler programs and the files it is working on). We use it to save space on our system device, which is a single floppy with a 400-odd OS/8 block capacity. Having to specify the library explicitly every time is a nuisance (and means the convenient CCL .LOAD command cannot be used.) The following patch to LOAD permits adopting a device other than SYS: as the default device for the FORTRAN library.

```
.GET DSK LOAD
.ODT
     13214/xxxx 5615                 JMP I .+1
     13215/xxxx 3636                 PATCH

     13626/xxxx 7200      PATCH,     CLA
                4777                 JMS I (200
                0012                 12
     13631/     0424        DN,      DEVICE DTA0
     13632/     0160
                0000                 0
                7402                 HLT
                7200                 CLA
                1232                 TAD DN+1
                6202                 CIF 0
                4773                 JMS (IOHAN
                1232                 TAD DN+1
                4777                 JMS I (200
                0002                 2
                3334                 LIBRY
                0000                 0
                5776                 JMP I (NOLIB
                1244                 TAD .-3
                3775                 DCA I (MODTBL+2
                1232                 TAD DN+1
     13652/xxxx 5774                 JMP I (3225

     13773/xxxx 0600
     13774/xxxx 3225
     13775/xxxx 6174
     13776/xxxx 3242
     13777/xxxx 0200
```

Note: the default device is in locations 13631-2 in DEVICE format, the default library filename is in 13334-7 in FILENAME format and is normally FORLIB.RL. The above code makes the default library DTA0:FORLIB.RL. The above code will work for any OS/8 device; it need not be co-resident with SYS:, etc.

A few notes from Ernst Lopes Cardozo.
Physiology Laboratory University Utrecht
Vondellaan 24, Utrecht, Holland

New MULTI8 developments

A number of improvements have been made to MULTI8 recently. Incremental plotters (XY8E) are fully supported now. Pen movements are accumulated by a plotter emulator task and writen into a FIFO disk file. Concurrently, the file is read by a subsidiary task and sent to the device. In this way the plotter can run at its own speed without slowing down the program or tying up a terminal.

A patchfile is available that makes Fortran IV dynamically addapt to MULTI8 background operation if necessary. An addapted version of the Fortran IV plotter routines (XYPLOT.RL) is available too.

The TC01/TC08 DECtape driver task has been improved. Special feature is the abillity to read PDP11 formatted tapes (256 words of 18 bits). Based on this new driver are two programs, DIR11 and COP11 that run in the MULTI8 background. With these programs it is possible to read and write RT11 files, and to obtain directory listings of RT11 tapes.

MULTI8 goes networking

Due to its flexible foreground task structure, MULTI8 is well suited to data communication and networking. Already one year in operation is a package (foreground task + background program) for remote job entry to CDC CYBER machines, emulating a CDC 200 User Terminal. Currently in development is a package that implements device-sharing among multiple MULTI8 machines. A pilot implementation was based on a simple hardware link consisting of two KL8E's (teletype interfaces) running at 9600 Baud. This version (now in operation) is extremely simple, yet allows the slave machine to access disks, tapes and lineprinter of the master. In fact the slave machine doesn't need to have a system device at all. After togling a 10 instruction bootstrap OS/8 comes up, equiped with disk, dectapes and lineprinter, all running on a machine with just memory and a terminal!

A more sophisticated protocol will be implemented that alows multiple datastreams running concurrently over one physical link. This protocol will be symetrical, i.e. there will be no fixed master/slave relation among the machines. So users on machine A may use peripherals on machine B and users on machine B may at the same time use peripherals on machine A.

The ultimate goal of this effort is loadsharing among multiple MULTI8 machines. Imagine a network of a number of PDP8's, all running MULTI8 with the networking software. The machines will probably have different configurations, with the DECtapes on one machine, lineprinter and magtape on a second machine, and maybe a floating point processor on a third one. Somewhere in the network a user starts his background program (running under control of OS/8 on his virtual machine). At some point his program starts to output to the lineprinter. The timesharing subsystem on his machine collects his data and sends it over the network to the machine where the lineprinter is. A moment later, our users' program starts the FPP. It does so by executing certain IOT's which are trapped by tha timesharing hardware. In this case the timesharing subsystem concludes that there is no FPP on this machine, and consequently ships the complete background (all his virtual memory plus registers etc.) through the network to the machine that has an FPP. The same mechanism may be used by the (distributed) Network Scheduler to distribute the load over the available processors. The user will never know where his program is executed. To him the system behaves as a timesharing system equiped with a large set of peripherals and a low load level. Clearly, such a system requires very fast DMA links to connect the processors.

There are two fundamental differences between DECNET and 8LINE. DECNET is designed to operate over long distances (telephone) at medium speeds. 8LINE is designed to operated over raliabel, fast or very fast connections which normally will be in-house, linking machines which are only a few hundred meters apart or even in the same cabinet. (Just think of the power of one cabinet with eight eight's !).

Over 32K memory

In the Las Vegas notes in Newsletter 20 Bob Hassinger wrote about discussions regarding the extension of the PDP8 memory capacity beyond 32K. Unfortunately, he did not give any detail about the "fairly sophisticated approach" that has been talked about. As I think this Newsletter is a good place to continue the discussion I will make a start by offering some of my own considerations. Reactions are firmly solicited!

First of all one should make a distinction between the memory size and the size of the address space. To enlarge the address space is a fairly involved operation requiring extensive reprogramming of programs that should make use of the extra memory. I don't think it can be done economically. In addition, there is hardly a need for a larger address space; only very large Fortran IV programs might benefit.

However, it is relatively easy to increase the maximum amount of memory that can be accomodated in the machine. It requires one or a few extra Exteded Memory Address lines (one reserved line is still available on the Omnibus) plus a means to generated the 4 or more bits wide EMA's. Lets assume for a moment that only one extra address line is added, giving a 64K capability. Because we don't increase the address space of the normal machine, the extra memory can only be used when the normal 32K address space is mapped on the available 64K.

Suppose we run a foreground/background system, with some form of memory management hardware that maps the virtual addresses generated by the processor (in User Mode) onto the 64K real memory. This requires that the mapping table of the memory management hardware produces 4 bit EMA's, which is no problem. Only the timesharing system (in MULTI8 the timesharing subsystem) has to be addapted to make use of the extra memory. The effect would be a much better real to virtual memory ratio, vastly increasing the systems capacity for large user programs. Of course, two little problem have to be solved:

At a few places the timesharing subsystem has to have access to the users memory, e.g. to inspect instructions etc. This can be solved in the memory management hardware too, by implementing an instruction CDFV. This instruction (Change DataField Virtual) loads a register in the memory management hardware and sets the interrupt inhibbit flipflop. After exsecution of this instruction each indirect memory reference is mapped to the real field loaded by the CDFV instruction. This mode of operation remains in effect until the next JMP or JMS, which also re-enables the interrupt system. Although rather baroque, this construction is sufficient to give the timesharing subsystem access to the users' memory.

The second problem deals with DMA devices. Clearly the use of the extra memory would be very restricted if it is not possible to transfer data between disk, dectape etc. and the new memory. At first sight, this requires addaptation of every DMA controler, or order to add a bit to the EMA register. There is however, a simpler solution. When a DMA transfer has to occur, the device controler will assert the Break Request line on the Omnibus. Then, during TS4 of the cycle preceding the actual break cycle, each active DMA device asserts a unique data line in order to sort out priorities. This fact can be used to determine which device is going to make the next break cycle. Once this is known, a simple piece of logic can drive the extra EMA line during the break cycle. So appart from loading the device control in the normal fassion, a device handler has to load a one bit register (presumably located in the memory management hardware), corresponding to the DMA priority of the device.

If another free line on the Omnibus can be found, the above schema can be extended to larger memory sizes without problems. The main point is the software addaptions are relative few, and all located in rather recent and modular programs (RTS8, MULTI8). There is no need to change a single bit in OS/8 or its language processors.

The constructions described above may even give a solution for the FPP, until now never supported in timesharing systems. The problem with the FPP is that it does not fit in the normal memory protection schema of the PDP8. However, with the descibed hardware it is possible to map all addresses generated by the FPP into some 32K section. (Remember the FPP is just another DMA device). This implies that the timesharing software has to place FPP jobs in that particular memory area.

This is certainly not the only feasable approach. I have the impression that DEC is looking (working?) in another direction, more or less along the following lines. Suppose you place all the memory modules on a special section of the Omnibus, which is driven through a converter that contains the necessary mapping functions. In fact it could contain one set of mapping registers for the processor, and one for each of the possible 13 DMA devices. This means that all devices that access memory are looking to it through the address translation logic. (c.f. the PDP11 memory management hardware, where only the processor looks through the mapping as it is the only device the generates addresses narrower than the actual memory address bus).

But once this construction has been made, you might want to go one step further and add a cache memory to the address translator. That would allow the processor to run at a much higher speed (e.g. 500 ns cycles). To remain compatible with the large set of existing interfaces, IOT cycles and DMA cycles should remain at 1.4 us (the destinction can "easily" be made by the timing generator).

Maybe all of this is pure nonsence. However, if none of these or similar things will be implemented, it won't be because of technical impossibilities. I hope that someone will pick up this monologue and make it a true discussion.

Login for OS/8

A nice LOGIN capability can easilly be build on the CHAIN RETURN patch submitted in Newsletter 19, page 14. Just write a program called LOGOUT.SV that does a lookup for it's own save file (which should be on SYS:) and enters this blocknumber in word 177 of block 0. If the system is bootstrapped, this word is loaded in word 17777 and as soon as the keyboard monitor comes in, he will chain to LOGOUT.SV. Now LOGOUT should perform a LOGIN operation, asking for user name and password etc. LOGOUT.SV can easily be written so as to make it impossible for the user to skip it, e.g. by patching location 7605 so that a manual restart at 7600 won't work. The only requirement is that the user should run LOGOUT when he leaves the system.

NOTE FROM BILL HAYGOOD

Bill should have ready by the end of July a new version of his multi-user OS/8 executive system which will support up to 4 terminals on a PDP-8/A/e/f/m with as little as 12K memory. Each terminal on the system has its own virtual 32K PDP-8 (even if the real system has less memory) and access to all disks (cartridge and floppy), paper tape reader/punch, line printer and EAE. (A one word patch to FORTRAN IV will allow it to run under MULTOS-8 using the EAE to get really fast computation.) Although the system is based on the DEC RK8-E as the main disk (supporting up to 4 drives), plans are being formed to allow use of the System Industries disk as the main swapping device. Other supported equipment includes memory to 32K and the RX8-E floppy disk (in either/both 12- bit or 8- bit modes). The system clock may be any one of the following: DK8-EA, DK8-EC, DK8-EP, or DKC8-AA. Future enhancements will include a special class of IOT's (no additional hardware required) to allow simulation of floating point operations. Interested parties may obtain additional information by writing to: W. F. Haygood, Jr., 3704 Ridgecrest Drive, Salt Lake City, UTAH 84118, telephone 801-966-1414.

FROM ANDREW SHORT

Another RESORC Table found!

This is an addition to the note by I.M.Templeton in Newsletter 20. DEC multiple-unit handlers have names of the form xyAn or xyBn, where x and y are letters and n is an octal digit, e.g. DTA0, RKB1. RESORC deciphers these by referring to a table of 7 xy pairs at 13756 to 13764 (DT,MT,LT,TD,CS,RK,RF). If it fails to find a match, it then refers to any three-letter names in the table at 14435 to 14466 and adds a digit in the fourth position.

For example, our non-DEC disk came with handlers named DSK0 and DSK1. RESORC called them DSK0 and MTB0. After 13757/MT was changed to DT, it named them correctly. However, I preferred to adopt the DEC convention by changing the handler names to DKA0 and DKA1, and changing 13764/RF to DK.

Incidentally, there is a bug in RESORC which gives false two-letter names (not recognized by OS/8) to some three- and four- letter combinations. These can be avoided by patching the RESORC tables with the new names.

THE UNIVERSITY of ROCHESTER
School of Medicine and Dentistry
Rochester, New York 14642

DEPARTMENT OF RADIATION BIOLOGY
AND BIOPHYSICS

May 6, 1977

Bob Hassinger
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, MA 01748

Dear Bob:

Everytime I read the wonderful 12 Bit S/G Newsletter I am prompted to write. However, the problems, solutions, heartbreak, and happiness take so long to write that the letter never gets finished.

Enclosed are very brief notes on many and related things. I hope some will save others time that we lost, others should generate discussion and/or answers. The RTS8 - floppy problem is our largest.

Kathi Albertini gave us the enclosed DECNET Report from the PDP10 - Decus meeting. A "BAT:" example is enclosed as there is little written about this.

Oh yes, we would very much like to hear about more memory. 32K is not enough! The writers of RTS/8 have spoiled us forever.

A writeup of DEC10, by Bob Phelps, is enclosed. It is very nice for asynchronous communication to the PDP10.

Please call if there are any questions about any of this stuff. (716-275-3791).

It might be a good idea to tell everyone again about Phelps USR routine for Fortran IV. I have been using it for years and could not do without it.

Yours,

Geoff

Geoffrey B. Inglis

GBL:rvb
enc.

BUGS - PROBLEMS AND QUESTIONS FOR OS/8 - RTS/8 USERS

(1) PAL8 V10A does not work properly in the background when core size is set to zero.

(2) Either RALF or PASS3 of Fortran IV does some strange things with "DSK:" - with "DSK" built to say RKA1:, an execute FOO.FO will create an "LD" file on "DSK" but will delete a corresponding FOO. LD on the system disk ("SYS:")

(3) BATCH will always hang on the LPT check in the background due to RTS/8 design - this looks like it will be fixed with V2B RTS/8.

(4) BATCH (the one distributed with OS/8 V3 (don't know about V3C) checks core by issuing the CDF 6XX1. In a 32K system it executes one too many 10T's - doing a 6301 (a user 10T). A 1705 board with a device select of 30 will have it's interrupt turned on - causing endless problems.

The above two problems have been fixed with a version of batch called BHAV V7.

(5) Speaking of 1705 boards, the older versions (REVS) of the 1705 have a defect in that data is strobed to the output at the same time as the data break priority bit is on the data lines. This gives low probability random noise on one to two of the bits in a 1705, according to the priority of the data break devices. This is very noticeable in a time sharing environment, like RTS/8.

(6) This is well-known but often forgotten problem--serial interfaces such as the KL8J--or older TTY 10 has the interrupt turned on with a clear from switches or CAF (6007) on 8/E. This interface is often used for line printers and one must remember to turn the interrupt enable flag off before using the LPT under RTS/8, if you use the kludge system of the LPT in OS/8 - RTS/8. (V2B should be the answer here--spooling will also be easy!).

(7) The equivalence statement of Fortran IV does not always work in 3.05. (It seems to in V 3.03 when equivalencing 3 dimension arrays.

(8) Writing in true 9 track format on a $12,000 TU10 (TM8E) on the PDP8 with OS/8 CUSPS, or the latest of the DEC handles is not possible. The data break controller makes this so. Expansion of the buffer is necessary for 9 track format--something which could have been done in hardware. Bob Phelps has written a handle which requires 16K but will work with FOTP PIP. Peter Lemkin's MAG10 did not work for us (Fortran II on the 8?) Our PDP10 sail must be different since all we got were error messages with the companion PDP10 version. The whole problem needs much more discussion in the newsletter.

BUGS - PROBLEMS AND QUESTIONS FOR OS/8 - RTS/8 USERS - continued.

(9) CCL is the bug of the year and my longtime gripe. V3C release is no better if not worse.

a) When ASSIGN is used to assign DSK: to other than the built DSK: the output side of a CCL command goes to the built DSK and the input side comes from the assigned DSK:. DSK should be looked up as in V3C's Direct to see if it is assigned. (The software support manual even suggests this.) Help should take files from the SYS: only.

b) Help does not need a special routine since it only need chain to direct.SV if the user types help or help' or chain to pip.sv if a file is specified.

c) CREF, for most uses need only call PAL8 with /c option set-- as distributed it doesn't work at all.

d) VERSION or some other command could return -- the CCL reminiscence. It is not difficult as mentioned in JAN Newsletter.

e) The tables don't match error states that things are incompatible. It should say "TYPE R CCL TO INSTALL". To new users incompatible means hopeless! (as in no TTY: handler in system).

f) All the above problems plus many small ones are fixed in a version from our lab called Version U.

Here are some problems we have not been able to solve:

(10) Under RS8/8 - OS/8 - 3 control C's in the buffer and nothing works for two or three CCL commands thereafter!

(11) With a task LPT: handler in V2B how will the SET LPT: command work?

(12) It would be nice if you could flag CTRL character in LPT: handler.

(13) The $64,000 RTS/8 question.

Since we use only field 0 for RTS/8 we often need to restore the <u>REAL</u> OS/8 field one monitor head before returning to OS/8. This is done by turning the trap off and calling the old OS/8 system field 0 handler to read block 0 to field one page 7600. This works fine for RK8, RK8E and tape, but not for floppies. WHY????

How might we do this if we can't read block zero with the floppy handler.

BUGS - PROBLEMS AND QUESTIONS FOR OS/8 - RTS/8 USERS - continued.

(14) It would be nice to be able to patch the system USR ENTER call to put in our time of day in an extra information word. Is there an easy way to do this? Think how fancy you could make CCL then, just like the PDP 10.

AMONG THE WISH YOU WOULD LIST!

(15) Reserve the free system block for keeping the background system head. A restore OS/8 background system head task is small, easy and very useful. It could be added to RTS/8 OS/8.

(16) We have also incorporated an unknown IOTLST for RTS/8 - OS/8 which does whatever the OS/8 IOT is supposed to do when in the list--this makes it easy to implement PTR, PTP--and even data break devices (with one more routine to take care of the field IOT). It saves many RTS/8 tasks and much work and space.

(17) Bob Phelps has written a routine for ASCII communication between PDP10 and 8/E--even in background--can talk at 9600 baud--with any speed TTY: full wild card functions of OS/8 command decoder. Works very well for ASCII.

Does anyone know of PALIO for the DEC10 which is up to date!???

(18) Hardware Notice!

Printronix makes a line printer (300 lines per minute) --with upper and lower case standard. It is very nice and high quality print on standard paper.

BUGS - PROBLEMS AND QUESTIONS FOR OS/8 - RTS/8 USERS - continued

(19) The Sked Users Group now distributes a time sharing state notation language which runs in the foreground - while OS/8 runs in background. A background monitor compiles loads and allows variables to be changed while as many as twelve different independent programs run in the foreground!

The language has limited mathematical capability and is designed as a process control and data collection system!

Data collected in foreground may be analyzed in background by Fortran IV, basic etc. while as many as 12 data files collect data! Clock resolution is 10ms. but can be run faster with fewer stations.

A manual and much more information is available from Dr. Snapper, SKED Users Group, Kalamazoo, Michigan; telephone (616) 383-1872 or contact G. Inglis, University of Rochester Medical Center, Department of Radiation Biology and Biophysics, 00-G-22, Rochester, New York 14642; telephone (716) 275-3791.

THE FOLLOWING IS AN EXAMPLE OF USING THE "BAT" HANDLER TO READ INPUT FROM THE BATCH FILE RATHER THAN THE THE KEYBOARD. THE BATCH HANDLER (BAT:) MUST BE INSERTED INTO THE OS/8 SYSTEM USING BUILD.SV THE HANDLER IS PROVIDED BY DEC IN VERSION 3 AND 3C OF OS/8. THE BAT HANDLER CAN BE CALLED BY ANY PROGRAM SUCH AS BASIC, FOCAL OR FORTRAN. IT WILL READ UP TO THE "$" . DATA IS THEN READ FROM THE BATCH BUFFER AS SPECIFIED BY THE FORMAT STATEMENT OF THE READ STATEMENT.

```
$JOB TO TEST BATCH HANDLER
.DA
.FRTS TEST
 HI--THE BATCH HANDLER WORKS!!
/THE $ BELOW MEANS EOF TO THE BAT: HANDLER
/THE BATCH HANDLER WILL EAT THESE COMMENTS
/AND EVERYTHING ELSE 'TILL THE "$".
$ DONE WITH ANSWERS
.DA
$END
```

```
  OS/8  FORTRAN IV  3.05              JAN 16  1977          PAGE  ONE

        C       TEST BATCH FORTRAN
0002            DIMENSION TEXT(5)
0003            DATA YES/'Y'/,UNIT/4/
0004            WRITE (4,29)
0005     29     FORMAT('OTAKE INPUT FROM BATCH FILE (Y OR N)  ?:',$)
0006            READ (4,25)ANSWER
0007     25     FORMAT(A1)
0010            IF (ANSWER.NE.YES)GO TO 60
0011            UNIT =9
0012            CALL USR(UNIT,'BAT:',2,ERR)
0013            IF(ERR.NE.0)WRITE (4,100)ERR
0014     100    FORMAT('USR ERROR ',I3)
0015     60     WRITE(4,20)
0016      20    FORMAT('OENTER TEXT')
0017            READ(UNIT,30)TEXT
0020      30    FORMAT(5A6)
0021            WRITE(4,50)TEXT
0022      50    FORMAT(1H ,5A6)
0023            STOP
0024            END
```

DECNET OVERVIEW
CHAIRMAN: PAUL HASHFIELD, ONLINE SYSTEMS, PA

NATHAN TEICHOLTZ FROM DEC GAVE AN OVERVIEW OF THE STATUS OF THE DECNET PROJECT ALL OF THE PROTOCOLS: DAP, NSP, NICE AND DDCMP ARE NOW STABILIZED. DAP MAKES SENSE OUT OF THE INFORMATION THAT FLOWS THROUGH THE NETWORK. NSP TAKES CARE OF ROUTING FLOW CONTROL, MAINTAINS LOGICAL LINKS, PROVIDES INTERRUPT MECHANISM AND PROVIDES ONLINE MAINTENANCE FEATURES. NICE DEALS WITH COMMAND AND ERROR INFORMATION. DDCMP IS THE PHYSICAL LINK PROTOCOL. THE NSP DOCUMENTATION HAS NOT YET BEEN CLEANED UP SO IT IS NOT YET AVAILABLE TO THE USERS. ANF10 WILL BE RELEASED WITH 6.03 AND PROVIDES A WAY TO MIGRATE TO DECNET BUT IT IS NOT DECNET PROTOCOL. JUDGING FROM USER COMMENTS AT THIS MEETING, THE NSP PROTOCOL HAS BEEN SOMEWHAT VOLATILE DURING ITS DEVELOPMENT. IT WAS STABILIZED PREVIOUSLY AND CHANGED FAIRLY RADICALLY SINCE THEN, SO PEOPLE PLANNING DECNET TYPE THINGS SHOULD PROBABLY BE AWARE OF THIS.

THE FOLLOWING CHART SHOWS THE STATUS OF THE DECNET SOFTWARE AS OF THE END OF 1977, SOME OF THESE FEATURES ALREADY EXIST. THE NUMBERS BELOW INDICATE THE MACHINE THAT THE FEATURE IS OR WILL BE IMPLEMENTED FOR. THE 10 IS NOTICEABLY LACKING IN FEATURES. THE USERS WERE ENCOURAGED TO MAKE KNOWN THEIR NEEDS FOR DECNET FLATURES TO BRIAN SAMUELS.

1	-	10
2	-	20
3	-	RSX 11M/S
4	-	RSX 11D IAS
5	-	RT11
6	-	RSTS/E
7	-	RTS 8

DECNET FUNCTIONALITY	1	2	3	4	5	6	7
USER/PROGRAMMER							
TASK TO TASK	X	X	X	X	X	X	X
FILE TRANSFER			X	X	X	X	
FILE ACCESS			X	X	X		
DATA ENTRY TERMINAL		X					
TIME SHARING TERM.		X					
COMMUNICATIONS							
POINT TO POINT	X	X	X	X	X	X	X
MULTIPOINT		X	V2		V2	V2	V2
ROUTING		X	V2				
GATEWAYS		V2					

V2 INDICATES THAT THE FEATURE WILL BE AVAILABLE WITH VERSION 2 OF THE DECNET SOFTWARE.

DEC10 runs on a PDP-8 and allows the user to communicate with a PDP-10 via the PDP-8's console terminal and a KL8-J asynchronous communications interface. The user can engage in two-way conversation with the PDP-10 as if the PDP-8's terminal were hooked directly to the PDP-10. In addition, the user can issue commands which transfer files between any OS/8 device and the PDP-10 disk.

Communication with the PDP-10 is initiated with the OS/8 command

```
.R DEC10
```

or the CCL command

```
.TEN
```

to which the PDP-10 will respond if it is connected and running. The next "." to appear on the terminal is a prompt from the PDP-10 and not from OS/8. At this point, the user can type any legal command to the PDP-10 and receive its reply as if the PDP-8 was not even present.

1.0 SPECIAL COMMANDS

If the user begins a command after the PDP-10 "." prompt with a "/", the command will be intercepted by the OS/8 command decoder and will result in any of several actions. The command must include one or more switches and an optional device:filename.ext specification. Since the "/" is transmitted to the command decoder, the command must begin with a switch specification. Other switches, of course, may appear anywhere in the command line and may be combined in any logical combination.

1.1 Login Switch -- /L=nn

The /L switch logs the user onto the PDP-10 account [1001,10nn]. The user must supply the password. If used in conjunction with other switches, the "/L" is superfluous if the equals option is non-zero. If /L is specified without an equals option, or if the equals option is zero, the PDP-10 is logged onto account [1001,1004] and the password is supplied by the PDP-8. The user may log onto any PDP-10 account with an explicit LOGIN command.

1.2 Logoff And Return Switches -- /O And /X

If either /O or /X is included in a command decoder specification, control will be returned to OS/8 after completion of the current command. /O will log the user off of the PDP-10 and /X will return to OS/8 without logging the user off.

1.3 Crash Character -- ^\

If the user gets stuck (for example if the PDP-10 crashes) and the normal commands don't work, control will be returned to OS/8 when the user types a CTRL-\ (control backslash, which on some terminals is a CTRL-SHIFT-L).

1.4 Help -- /H

For a summary of commands and switches, type "/H".

2.0 FILE TRANSMISSION -- /S AND /R

Files may be transmitted to or from the PDP-10. The basic command is

```
     /S dev:filename.ext
```

to send files to the PDP-10 and

```
     /R dev:filename.ext
```

to receive files from the PDP-10. For this form of the command, the files must have the same name on the PDP-10 and the PDP-8. Only one file can be received with one command, although it may be the concatination of several PDP-10 files (see the /N command below). Several files may be sent with one command, however. In this case, the filenames and devices are separated with commas using standard OS/8 conventions and the "wild card" characters "?" and "*" are allowed. If no OS/8 device is stated, DSK: is assumed and if no extension is stated, .XO is assumed.

2.1 Explicit PDP-10 Filenames -- /N

Since OS/8 only allows 2 character extensions, explicit names must be specified to transfer PDP-10 files with three character extensions. If the /N option is specified, the DEC10 program will prompt the user for the PDP-10 filename.

This feature may also be used to specify multiple PDP-10 files to be concatinated upon reception into one OS/8 file or to specify ppn's other than the one the user is logged onto. A maximum of 24 characters is allowed in the specification.

2.2 Binary Files -- /B

Addition of the /B switch to a command specification will result in special binary format files being transmitted. Files transmitted from the PDP-10 are assumed to be in "image" format such as those created by the assembler PAL10. They will be stored on the PDP-8 in a format accepted by ABSLDR. Files transmitted from the PDP-8 are assumed to be double precision (24 bit) integer numbers and are stored on the PDP-10 in unformatted FORTRAN format. A 256 word OS/8 block can be read with a single unformatted READ statement into a 128 element list. The second 12 bit word of each pair on the PDP-8 is assumed to contain the high order bits. The two 12 bit words packed into each list element can be separated simply by dividing by 4096.

2.3 Execute --/E

It is possible to send a file to be executed as a PDP-10 command rather than having the file stored on the disk. To do this, use the /E switch rather than the /S switch. Caution: Since the entire file is sent without 10, the file must not be longer than the PDP-10 will allow the user to type ahead.

3.0 USING DEC10 WITH PDP-8 CCL AND BATCH

All of the commands may be specified in a single command line with the CCL command TEN. When a legal file transmission command is specified as part of the CCL command TEN, control always returns to OS/8 and the user is logged off unless the /X switch was specified. This feature can be used to send files and submit jobs to the PDP-10 under OS/8 batch. For example:

```
        .TEN DATA.10/S/B/L/X      /Login and send a data
                                  /file, binary mode, to the PDP-10.
                                  /Don't logout when done.
        .TEN SU10/E               /Submit a batch job to the
                                  /PDP-10 to process the data.
                                  /SU10.X0 contains the statement
                                  /  .SU filename
                                  /Logoff when submitted.
```

Heinz Stegbauer
HTBL - Moedling
Technikerstr. 1 - 5
A - 2340 MOEDLING
Austria

Mödling, 77-06-02

Robert Hassinger, 12 Bit SIG Coordinator
c/o DECUS
146 Main Street
MAYNARD, MA 01754
USA

Dear Sir,

having drawn a lot of good ideas from the 12 Bit SIG Newsletter till now I'm glad to be able to contribute a little for my part. Please, make use of the enclosed article if you think it might interest anyone.

I am a teacher at a technical college near Vienna. We own a PDP-8/e with 28K of core memory, TCØ8 & 4 DECtape drives, 3 TTY's, 3 LA36 and a plotter. Normally we run EDU-25 BASIC, sometimes OS/8 for demonstration purposes. For plotter programs we use OS/8 BASIC with a selfmade user function package.

Yours sincerely Heinz Stegbauer

ONE DECTAPE EDU-25 BASIC

Referring to the contribution of Mr. Olson in the issue of March (No. 21) concerning EDU-25 BASIC usage with a single DECtape drive I want to remark:

The proposed solution - ASSIGNing DTA1 to SYS - is not possible since EDU-25 as a multiuser system is fully interrupt driven and therefore has its own internal handler indeed. EDU-25 uses DECtape Unit Ø for program storage and Unit 1 processing data files under program control. It's the handler to decide which unit to use by checking the filename extensions (.Ex or .Dx).

Changing location 13Ø43 from 1Ø76 to 7ØØØ (Version 3) forces the handler to direct all types of files to Unit Ø. This change doesn't affect the execution speed, except the processing of data files if another user is requesting program storage or retrieval operations concurrently.

However, renouncing to the use of data files in BASIC programs and commands like FILELOG, it is possible to run EDU-25 without any modification at all on a single DECtape drive.

MONTREAL NEUROLOGICAL HOSPITAL

AND

MONTREAL NEUROLOGICAL INSTITUTE

McGILL UNIVERSITY

3801 UNIVERSITY STREET - MONTREAL, CANADA H3A 2B4
tel: 842-1231, ext. 1841

Montréal, June 7th 1977

Mr.R. Hassinger,
Liberty Mutual Research Center,
71 Frankland Road,
Hopkington, MA 01748,
U.S.A.

Dear Bob,

I was interested in the recent discussion of high level languages available under OS/8. While I have never used Basic or Focal, I have used Fortran II extensively and recently used Fortran IV.

I have done a lot of programing in Fortran II using SABR subroutines to speed up certain calculations or for real time data collection. Most of our work is in EEG analysis and cerebral blood flow measurement and because of speed and data thruput requirements we use mostly integer arithmetic. We have a PEC magnetic tape unit and RK8 disk on a 32K PDP-12. We use the DECUS BLOCK I/O and a tape read/write routine that uses DMA transfers to the magnetic tape drive.

We recently bought a second-hand FPP-12. It was not until we started doing some Fortran IV programing that we realized a problem not mentioned by your other correspondents. Because the FPP-12 uses 15 bit addresses, it does not care about 4K memory field boundaries. This allows it to put arrays anywhere in memory, including across field boundaries. Since the disk and tape drives we have used independent address registers and extension field registers for DMA access, it is not possible for them to transfer data across a field boundary. This and the fact that we now have a great number of SABR routines means we have made very little use of Fortran IV except for off-line "number crunching".

In order to get some use out of the FPP-12, I have recently converted all the floating point routines on the Fortran II system to use the FPP-12 hardware. To do this it is necessary to modify the compiler to generate constants compatible with the FPP-12 and this will be done shortly. A speed factor of 6 is typical in loops using all real arithmetic for tests so far. So far only the basic functions "+" "-" "÷" and "x" use the hardware but selected functions, eg LOG or SIN could be rewritten to run entirely in the FPP-12 which would speed up programs which use these even more. I would be interested to hear from others who have tried something similar.

I have also modified the Format routines to allow commas to separate numbers on input I,F or E formats.

C.J. Thompson,
Computer Engineer

CJT/gr

June 8, 1977

Robert Hassinger, Coordinator
12-Bit S.I.G.
Liberty Mutual Research Center
71 Franklin Road
Hopkintown, Massachusetts 01748

Dear Bob:

This is just a short note in appreciation for, and in support of, the position taken by Lars Palmer in his communication which appeared in #22 of this newsletter. It's not that I don't appreciate the problems which confront users who have small systems, with sparse peripherals, and limited dollar resources, because I certainly do! (I've been there...) However, it often seems as if an inordinate amount of effort is being devoted to the development of languages and interactive-use systems (such as U/W FOCAL) for these limited configurations. At some point, there is surely a crossover between the yield to be expected from programming effort, and that to be expected from the expenditure of some resources on hardware.

We, too, have found that the expense of purchasing more core, some disks, an F.P.P., and the DEC-supplied high-level software, is less than that which would have been required to get a smaller system to do what we need by developing software.

Of course, I can only second Lars' comments about the tradeoffs involved in adopting systems such as U/W FOCAL. Our real-time processing needs, and our occasional need for super-fast number crunching in the middle of a real-time application, absolutely preclude the use of anything as slow as FOCAL. And, as was said, with the right hardware, using TECO or a similar editor, and CCL-controlled compile/execute sequences, OS-8 appears almost as interactive as a core-resident system. That is not to say that there isn't a lot of work to be done in interactive Program Development Systems (because there is), but just that the tack taken in U/W FOCAL (or in the BASIC executives) is only trivially helpful, compared to what is really needed.

Often, it seems as if the 12-bit group is really composed of two subgroups, and perhaps people-resources versus hardware-resources is a good way to define the division. We all appreciate the work done by those with people-resources to spare, but let's also keep up the good work and the information interchange among those with larger systems, who are using more of the high-level systems and languages.

Sincerely yours,

Dick

Dick Meltzer
General Electric Co. / Information Systems Programs
Arlington, Virginia 22202

GENERAL LATEX AND CHEMICAL CORPORATION (OF OHIO)

P. O. BOX 498 • ASHLAND • OHIO 44805
419-289-2727

May 3, 1977

Mr. Bob Hassinger
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, Massachusetts 01748

Subject: Request for elongated lines for listings (12 BIT SIG NEWSLETTER #21, page 6, par. 3)

Dear Mr. Hassinger:

I have implemented elongated heading lines for a Centronics printer. The program is a significant modification to CREF 12 (for PDP-12 DIAL-MS cross reference listings - source and binary in DECUS 12-208). This program is somewhat more involved than Jim VanZee's request - but it at least illustrates a method of elongating the title line of a listing. (I have not attempted changes to PAL 8 or PAL 12 yet.)

You immediately run into problems on elongation of characters - what if the listing is to a device (LPT or TTY) that accepts less than 132 columns? Since the elongation effectively doubles the column output you have to check "something" to be sure elongation is not going to print off page or worse - wrap around. In DIAL this was fairly simple (understatement?) and in OS/8, maybe looking up "columns" in the handler can be done in a straight forward manner.

I have sent three pages of the CREFTITL listing, the title page and, where the header is constructed. Let me explain a little more:

1. Depending upon "paper size", a pointer to the header buffer is set up.

   = 132 columns points to TITLE 2.
   < 132 columns points to TITLE 1.
   > 132 columns changes loc 2617 to 216 (elongate).

2. The date in DIAL is input within the CREF program - there's no dateword in DIAL.

3. The first line of a listing, if a comment line (starting with a slash) causes the first 8 characters or less to be the program name (or less if a space is found.)

RECOMMENDATIONS FOR THE USE OF OUR MATERIALS ARE BASED UPON LABORATORY TESTS AND EVALUATIONS BELIEVED TO BE RELIABLE. HOWEVER, THERE IS NO EXPRESSED OR IMPLIED WARRANTY AS TO RESULTS OBTAINED OR TO BE OBTAINED BY OTHERS WHO MAY MAKE USE OF THIS INFORMATION OR WITH RESPECT TO THE ABSENCE, EXISTENCE, OR VALIDITY OF PATENTS RIGHTS, IF ANY, OF OTHERS INVOLVING ANY COMPOSITION OR PROCESS HEREIN REFERRED TO; OR AN INDUCEMENT OR RECOMMENDATION FOR THE VIOLATION OF ANY SUCH PATENT RIGHTS, AND RESPONSIBILITY AND LIABILITY THEREFORE IS DISCLAIMED.

3. continued -

   These characters are also used to form a 10x12 character matrex for the big title page preceding the listing (similar to DEC 10 listings).

4. Up to 32 characters are used as the title line of each page.

5. The title line is completed by an octal page number.

If you or Jim would like more information on this program (and don't have DECUS 12-208), let me know - I'd be glad to send it along.

(Please excuse the Xerox copy paste-ups - I don't have the luxury of reduction copying).

Subject: Receipt of Newsletters.

Bob, I received the March Newsletter on April 28, one day before the deadline for material to be included in the May Newsletter. The January copy was received on March 2, etc. Therefore, I too must "complain " about the turnaround time.

Incidentally, the Newsletter is my only source of information re DEC. For some reason the Software News and OS/8 updates stopped coming in September 1976, and DECUS updates don't seem to come at all. Software ordered on LINCTAPE from DECUS has been (bluntly put) lousy, with tapes containing multiple errors, apparently introduced during copying. And you are never sure which version is being sent. May I suggest version numbers for DECUS submissions and that they be "written" somewhere?

Sincerely yours,

Sheldon

H. S. Hopkins, Jr.

HSH/ef

VG Data Systems

VG Data Systems Limited
Tudor Road, Altrincham, Cheshire WA14 5RZ, England. Telephone: 061-928 9747 Telex: 667079

9th May, 1977

Mr. Bob Hassinger,
12 Bit SIG,
c/o DECUS,
146 Main Street,
Maynard, MA 01754,
U.S.A.

Dear Bob

PDP8A Power Fail Protect

We manufacture real-time computer systems for scientific instrumentation, such as mass spectrometers and electron spectrometers, and have recently been reconfiguring our systems from PDP8E based to PDP8A/420 and PDP8A/620 based systems. This changeover has gone very smoothly after the expected initial problems of mechanical mounting of the CPU'S in our cabinets. On the software side we have been delighted since all of our software suite of 150K PAL-8 Assembler code (running under OS8/RTS8) executes without any modification, save for the insertion of the single instruction 6103 in every interrupt handler chain. Since it was not at all obvious that this would be needed I thought it would be worthwhile to document this problem for other PDP8A users.

The DEC-supplied hardware of one of our typical systems (of which we sell 50 a year) has 16K of core, RK8E disk, RX8A floppy and VT55 graphic display terminal. Translating this to the PDP8A, we of course need the KM8-A Option Board (M8317) to use the memory extension and time-share control facility. However this board also has, for free, the power-fail protect logic on it. Beware! This logic has an interrupt enabled flag, which cannot be disabled by links or switches. Furthermore the flag is set by the $\overline{\text{AC LOW}}$ signal rather than by reduction of the low voltage DC (+5 volt etc.). In our Industrial environment we have mains supply fluctuations and noise which have caused the following phenomena:-

1. AC LOW trigger of Power Fail :- Every 20 minutes (average)
2. Disk transport starts to run down:- 4 times in 6 months.
3. CPU RUN light goes out :- Once in 6 months.

Case 1 can be overcome by clearing the power-fail protect flag (CAL = 6103)
Case 2 just suspends operation of the system for twenty seconds
Case 3 is the one where power-fail protect software would have been useful.

Note that neither the PDP8A front-panel INIT button nor the software CAF instruction (6007) clears the power fail flag. (The PDP8A handbook does not make this point).

continued overleaf:-

Directors: [redacted], B. A. Eastwell, B.Sc., P. Powers, M.Sc., B.Sc., M. J. Perks, B.Sc., H. G. Streets, M. J. Wellington, Ph.D., B.Sc.

Registered in England No. 1139724.
Registered Office: Imberhorne Lane, East Grinstead, Sussex.

It also seems to me that RTS-8 V1 and 2 do not handle the power-fail logic for a PDP8A. Apart from assuming that if you have the power-fail protect facility you use it (as on a PDP8E), there is no CAL instruction (6103) which you need to clear the AC LOW interrupt (PDP8A Users Manual p.6-52). I understand that PDP8A support is being given in RTS-8 V.2B so no doubt the extra code will be added.

Yours sincerely

Mike Wallington

M. J. Wallington
Software Director

Copies to:- Mr. Gary Cole, DEC Maynard.
Mr. Derek Wood, PDP-8 OEM Support, Reading.

Query from Alan Cleary. New Castle Upon Tyne.

We have a 16k 8/E with one RK8E drive and DECtape, We have purchased a DP8E synchronous modern adaptor. with the intention of emulatin a 2780 RJE to transmit OS18 files to and from the University's IBM370. I have obtained the DECUS program 8-733, but it won't run under OS/8 and needs a card reader. Before starting to reinvent the wheel. I would like to know whether any other SIG member has any useful experience along these lines.

Remarks by Lars Palmer:
I have also thought along these lines and I'm quite certain that an answer to this has a large generel interest.

Hutton + Rostron, a firm of consulting architects in the UK, are setting up an information service dealing with software for the construction industry. This will complement their International Directory of Computer Programs for the Building Industry, which is being revised for publication later in 1977

The Directory is a world-wide source of reference for computer services to the construction industry, and covers all topics likely to be of interest to the architect, engineer or planner, from design of off-shore structures to irrigation planning, as well as subjects more closely related to the building industry, such as structural analysis, heating and ventilation, and design of electrical services

Information is collected by questionnaires, and abstracts are prepared and published as a free service. Hutton + Rostron will also prepare two-page descriptions of programs or services, to be included in a separate section of the book. These descriptions are prepared in the same way as those completed by the firm for the Department of the Environment and the magazine Building in the UK. As well as giving good publicity, these descriptions also provide a good basis for future company literature, which Hutton + Rostron will also organise if required

For further details, contact Hutton + Rostron, Netley House,
Gomshall, Surrey, GU5 9QA, UK